HTML5 for the Silverlight Guy

An introduction to “Serious” development on the Web Platform

David Padbury
http://davidpadbury.com
@davidpadbury

So, this whole Silverlight vs HTML5 has been a little controversial...

"Microsoft execs said tonight ... allow them to create Windows 8 applications in HTML and/or JavaScript.
Sinofsky and others didn't mention Silverlight or XNA at all."

http://www.zdnet.com/blog/microsoft/windows-8-more-than-just-windows-phone-on-your-pc/9592?tag=mantle_skin;content

‹ Previous Thread | Next Thread ›

Windows 8 apps going html5, wtf

192 replies

Last post Jun 03, 2011 07:23 PM by Psychlist1972

★★★★★ (5)

bitdisaster
Participant
1004 Points
263 Posts

Windows 8 apps going html5, wtf

Jun 02, 2011 12:06 AM | LINK

Just watched the first official Windows 8 video http://www.youtube.com/watch?v=p92QfWOw88I. While I like the UI/UX, I don't like that we are supposed to write Windows 8 apps in html/js. Wow, thats the stupids thing I ever heard. Microsoft has a first class cross-platform application framework called Silverlight and they want us to write freaking javascript. Really!? Probably the next version of WindowsPhone will run HTML5 as well. C'mon, I'm really disappointed.

Any thoughts?

Did I mention that i'm disappointed. HTML for text, XAML for apps, what's so complicated?

An disappointed Silverlight developer.

Jan Hannemann
Research Associate University of Victoria

Please remember to mark the replies as answers if they answered your question

Reply

eclipsoft
Member
44 Points
38 Posts

Re: Windows 8 apps going html5, wtf

Jun 02, 2011 12:56 AM | LINK

Potentially TERRIBLE news. It almost puts me in a state of shock. My biggest fears coming into Windows 8 was that, as a mostly WPF+.NET developer, was that they would shift everything to Silverlight and leave the FULL platform (can you write a VisualStudio in Silverlight? of course not, not designed for that) in the dust. To my utter shock, they did something much, much, much worse.

I say 'potentially' because, it is possible WPF and Silverlight will be able to be everywhere and do everything that the Html5+JavaScarry "platform" (replacing XAML + C#.... can you believe your eyes? I can't) can do in full integration into the

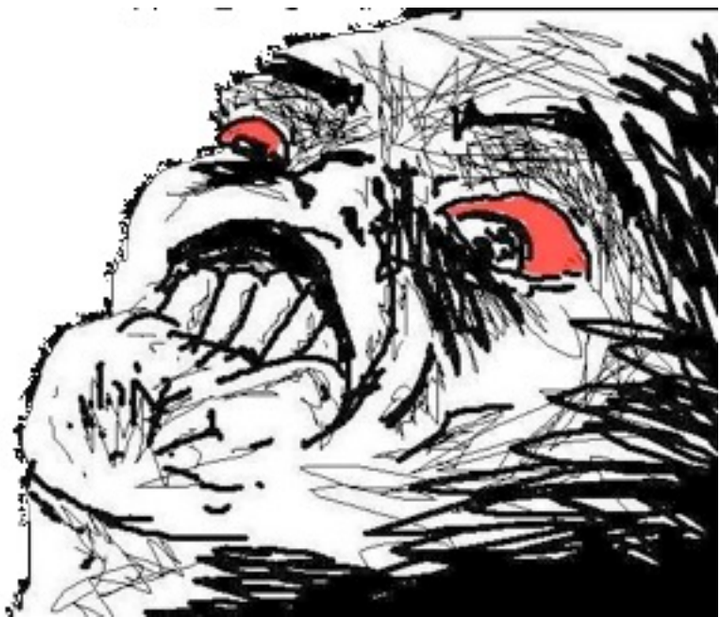
...Now they're not just f*****g Silverlight developers, they're f*****g WPF developers as well? ...

Just so there's no confusion, when my text is edited, the *** stands for

F - * - * - * - * - * - G

Because we've just been F****D.

http://forums.silverlight.net/forums/p/230502/562113.aspx#562081



“...they want us to write freaking javascript. Really!?”

“HTML5??? Microsoft isn't eating their own dogfood, they're eating their own vomit.”

“It may just be a very cool thing if they are brewing a way to utilize html5 + javascript **without us having to mess with javascript** (exactly what ASP.NET does on the client side)”

“At this point html5/js is a horrible development platform... I have not seen a single project developed purely in html/js.”

“we get Html and JavaScarriness in place of the beauty we know”

“Seriously, who decided that we should start basing all of our computing around the most abysmally slow language!?”

“Contrary to popular belief, HTML 5 is not ready yet.”

“there is no way you would write a trading application in HTML5 and javascript”

Javascript may be the WORST language ever used to write "applications". It's a joke.

▲ xradionut 3 hours ago | link

Yeah, let's have a richly, wonderful, HTML interface that still can't replicate desktop GUI productivity from the 90's...

reply

▲ WayneDB 3 hours ago | link

This is exactly the problem. HTML/CSS/JS is disgusting to someone who knows the power of a good native SDK.

People who only do web dev just don't know what quality is.

reply

"You can't write serious applications in HTML and JavaScript."

There's lots of cool HTML5 demos out there

But how do I build a rich front end application?

That's what I'm going to (try) helping you understand today.

But first.

Silverlight is not dead.

Evaluate HTML5 the same way you would anything else.

It's good at some things, it sucks at others.

Although this is the web,
forget your server side web programming experience.

ASP.NET, Rails, Grails, PHP

Well, not entirely but you'll see where I'm going.
I'm trying to be dramatic.

Writing complex front-end in HTML+JavaScript has more in common with WPF/SL than ASP.NET

For you guys, this is a good thing.

Let’s talk business.

Stock Trader Reference Implementation

CFI STOCKTRADER

POSITION | WATCH LIST

Add to Watch List

Symbol	Shares	Last	Cost Basis	Market Value	Gain/Loss %	Actions
STOCK0	10	$317.11	$280.99	$3,171.06	1028.5%	
STOCK2	100	$224.83	$1,900.22	$22,483.09	1083.2%	
STOCK3	100	$481.10	$1,900.22	$48,109.67	2431.8%	
STOCK6	50	$475.88	$523.43	$23,794.09	4445.8%	
STOCK7	25	$367.62	$6,990.13	$9,190.45	31.5%	

STOCK2 HISTORICAL DATA

PIE CHART

NEWS ARTICLES

3/31/2008 12:00:00 AM
Great Lakes Goes Healthy

12/6/2007 12:00:00 AM
Ravioli Scare Prompts Government Action

5/24/2006 12:00:00 AM
It's Pasta La Vista for Great Lakes

2/14/2005 12:00:00 AM
Henderson: It's Gone Well Pasta Joke

Application is built from modules

Modules are decoupled from each other

Code is easy to test

Runs on Desktop, RIA, and (windows 7) Phone

Today's Talk

JavaScript

JavaScript Modules
Module Organization
Views as Modules

Decoupling and Testing

Event Aggregator/PubSub
Unit Testing View Modules

CSS

CSS Compilers
Platform Consistency

HTML5 Everywhere

Plugins
Devices

We're going to talk about the *boring* bits of HTML5

"JavaScript doesn't suck.
You're just doing it wrong."

- Douglas Crockford
(father of JSON and JSLint, author of The Good Parts)

The tone's a little harsh, but he has a point.
Despite looking similar to C#, JavaScript is a very different language.

JavaScript is a very simple language

If you can learn C#, you can cope with JavaScript.

There are three things you have to learn to be a great JavaScript developer.

First Class Functions

Prototypes

Context

First Class Functions

```
function createCounter() {
    var count = 0;

    return function() {
        return count++;
    };
}

var counter1 = createCounter(),
    counter2 = createCounter();

console.log( counter1() ); // 0
console.log( counter1() ); // 1
console.log( counter2() ); // 0
console.log( counter1() ); // 2
```

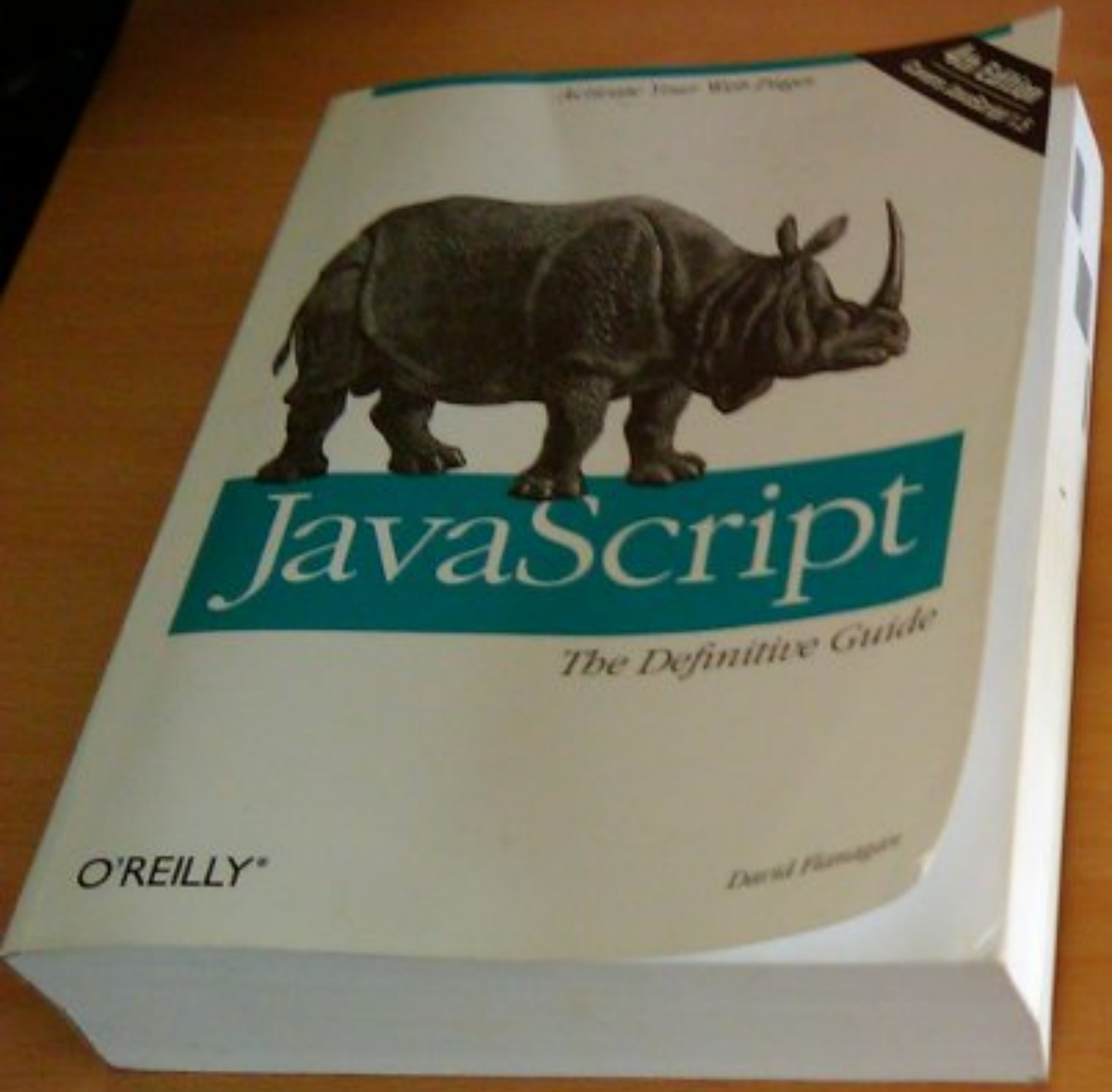
Context

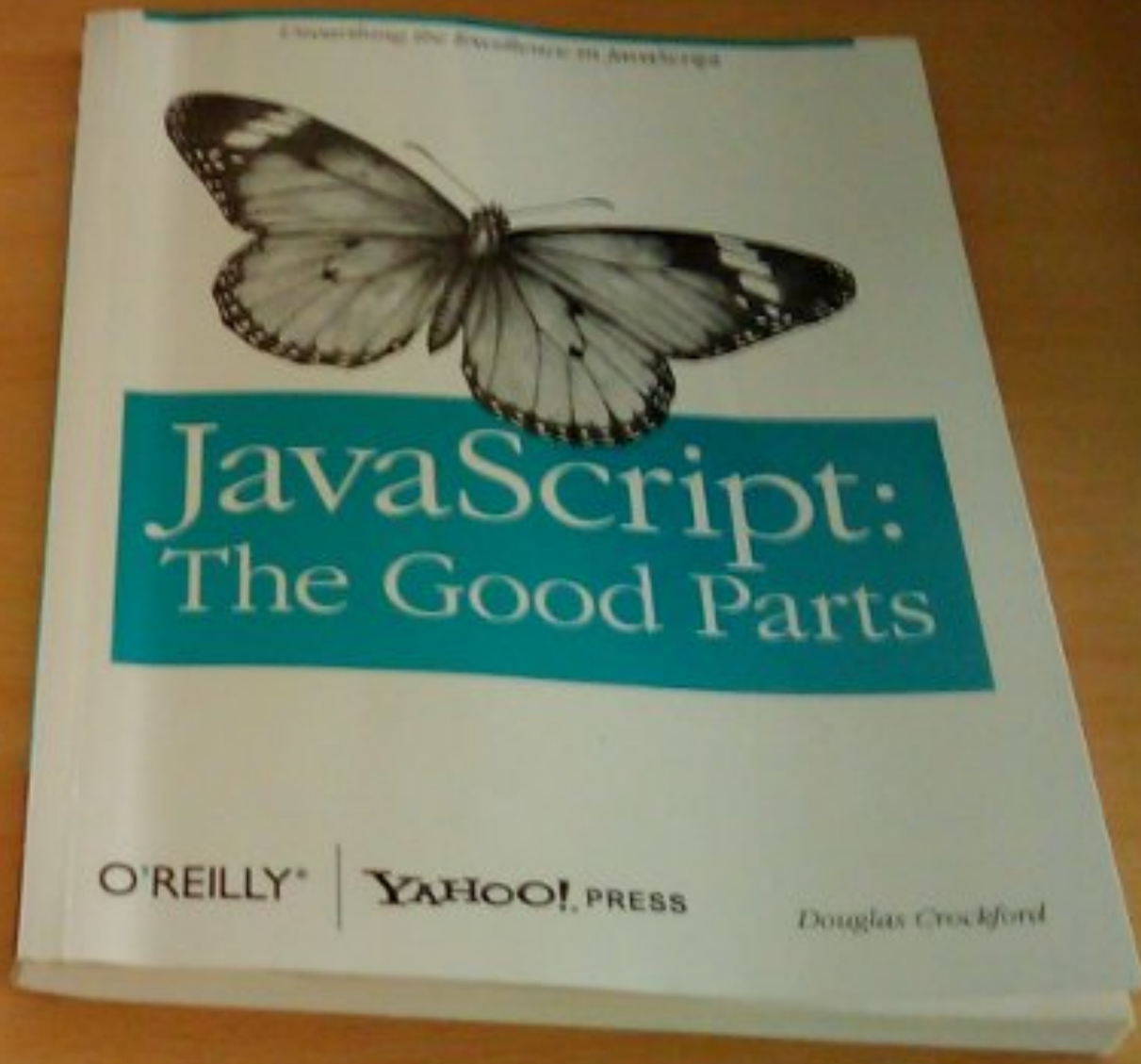
Or, which `this` is `this`?

```
var david = {
    name: 'David',
    sayHi: function() {
        console.log("Hi, I'm " + this.name);
    }
};

david.sayHi(); // Hi, I'm David
// obj.fn() is eqv to obj.fn.call(obj)

david.sayHi.call({name:'Colleen'});
// Hi, I'm Colleen
```

Prototypes

```
function Person(name) {
    this.name = name;
}

Person.prototype.sayHi = function() {
    console.log("Hi, I'm " + this.name);
};

var david = new Person('David'),
    colleen = new Person('Colleen');

david.sayHi();  // Hi, I'm David
colleen.sayHi();  // Hi, I'm Colleen

console.log( david.sayHi === colleen.sayHi ); // true
```





Some older bits aren't great...

`strict` mode keeps us on the Good Parts

Start a file with `'use strict';`

```
'use strict';

alert( "I'm in strict mode." );
```

Or start a function with `'use strict';`

```
(function() {
    'use strict';

    alert("I'm in strict mode!");
})();
```

```
(function() {
    globalVariable = 'OOPS!';
})();

console.log(globalVariable);
// OOPS!
```

```
(function() {
    'use strict';
    globalVariable = 'OOPS!';
})();

console.log(globalVariable);
```

```
ReferenceError: globalVariable is not defined
```

```
var person;

(function() {
    person = {
        name: 'David',
        name: 'Colleen'
    };
})();

console.log(person.name);
// Colleen
```

```
var person;

(function() {
    'use strict';
    person = {
        name: 'David',
        name: 'Colleen'
    };
})();

console.log(person.name);
```

SyntaxError: Duplicate data property in object literal not allowed in strict mode

```
(function() {
    console.log( 012 + 2 );
})();
```

```
(function() {
    console.log( 012 + 2 );  // 12?!
})();
```

```
(function() {
    'use strict';
    console.log( 012 + 2 );
})();
```

SyntaxError: Octal literals are not allowed in strict mode.

In some ways, too simple.

C# has `using`

Visual Basic has `Imports`

JavaScript has?

C# has `using`

Visual Basic has `Imports`

JavaScript has?
Nothing.

```
<!DOCTYPE HTML>
<html>
    <body>
        <script>
            // Application goes here...
        </script>
    </body>
</html>
```

```
$(document).ready(function() {
    function addSymbol(symbol) {
        var row = $('<tr>').append($('<td>').text(symbol));
        $('.watches').children().append(row);
    }
    $('.screen-switcher a').click(function() {
        var screen = $(this).attr('data-screen-id');
        $('.screens .screen').slideUp('fast', function() {
            $('.screens .screen[data-screen=' + screen + ']').slideDown();
        });
    });
    $('.add-symbol').parent('form').submit(function(e) {
        e.preventDefault();
        var symbol = $('.add-symbol').val();
        addSymbol(symbol);
        $.ajax('/data/symbol' + symbol, {
            completed: function(xhr, data) {
                $('<div class="price">')
                    .text(data.price)
                    .click(function() {
                        $('.price .tooltip')
                            .show('fast', function() {
                                $(this).text(price);
                            })
                    });
            }
        });
});
$('.sidebar .history').flot({
    data: $.ajax('/stock/history', {
        data: [1,23,4,5,6],
        date: new Date().getTime()
    });
```

Everything is global by default

```
// lib1.js
function something() {
    console.log('foo');
}
```

```
// lib2.js
function something() {
    console.log('bar');
}
```

```
<script src="lib1.js"></script>
<script src="lib2.js"></script>
<script>
    something(); // bar
</script>
```

The patterns and tools and practices that will form the foundation of Modern JavaScript are going to have to come from outside implementations of the language itself

- Rebecca Murphey

Using simple JavaScript constructs we can emulate many traditional organization techniques

```
(function(lab49) {

    function privateAdder(n1, n2) {
        return n1 + n2;
    }

    lab49.add = function(n1, n2) {
        return privateAdder(n1, n2);
    };

})(window.lab49 = window.lab49 || {});
```

```
lab49.add(2, 3);
```

Known as the Module Pattern

http://blog.davidpadbury.com/2011/08/21/javascript-modules/

Using simple JavaScript constructs we can emulate many traditional organization techniques

```
(function(lab49) {

    function privateAdder(n1, n2) {
        return n1 + n2;
    }

    lab49.add = function(n1, n2) {
        return privateAdder(n1, n2);
    };

})(window.lab49 = window.lab49 || {});
```

```
lab49.add(2, 3);
```

Known as the Module Pattern

http://blog.davidpadbury.com/2011/08/21/javascript-modules/

Using simple JavaScript constructs we can emulate many traditional organization techniques

```
(function(lab49) {

    function privateAdder(n1, n2) {
        return n1 + n2;
    }

    lab49.add = function(n1, n2) {
        return privateAdder(n1, n2);
    };

})(window.lab49 = window.lab49 || {});
```

```
lab49.add(2, 3);
```

Known as the Module Pattern

http://blog.davidpadbury.com/2011/08/21/javascript-modules/

Using simple JavaScript constructs we can emulate many traditional organization techniques

```
(function(lab49) {

    function privateAdder(n1, n2) {
        return n1 + n2;
    }

    lab49.add = function(n1, n2) {
        return privateAdder(n1, n2);
    };

})(window.lab49 = window.lab49 || {});
```

```
lab49.add(2, 3);
```

Known as the Module Pattern

http://blog.davidpadbury.com/2011/08/21/javascript-modules/

Modules have evolved into standard patterns with tooling to help loading and packaging.

Asynchronous Module Definition (Commonly known as AMD)

https://github.com/amdjs/amdjs-api/wiki/AMD

`define` for creating a module definition.

`require` to ask for a module instance.

```
define('calculator', function() {
    return {
        add: function(n1, n2) {
            return n1 + n2;
        }
    };
});

require(['calculator'], function(calculator) {
    console.log( calculator.add(1, 2) ); // 3
});
```

Callback so modules can be loaded asynchronously

```
define('math/adder', function() {
    return {
        add: function(n1, n2) {
            return n1 + n2;
        }
    }
});
```

Specify modules this module depends on

```
define('math/calculator', ['./adder'], function(adder) {
    return {
        add: function(n1, n2) {
            return adder.add(n1, n2);
        }
    };
});
```

Loader passes instances of them

Loaders can assume module names from files

```
// math/adder.js
define(function() {
    return {
        add: function(n1, n2) {
            return n1 + n2;
        }
    }
});
```

```
// math/calculator.js
define(['./adder'], function(adder) {
    return {
        add: function(n1, n2) {
            return adder.add(n1, n2);
        }
    };
});
```

AMD appears to be winning in how to large JavaScript code bases

There are a number of AMD module loaders

RequireJS, curl.js

Popular libraries are supporting AMD

Dojo, jQuery (as of 1.7)

Demo

Basic RequireJS

You can extend how modules are loaded using *plugins*

```
text!
i18n!
order!
```

```
// views/personView.html
<div>My name is: <span class="name"></span></div>
```

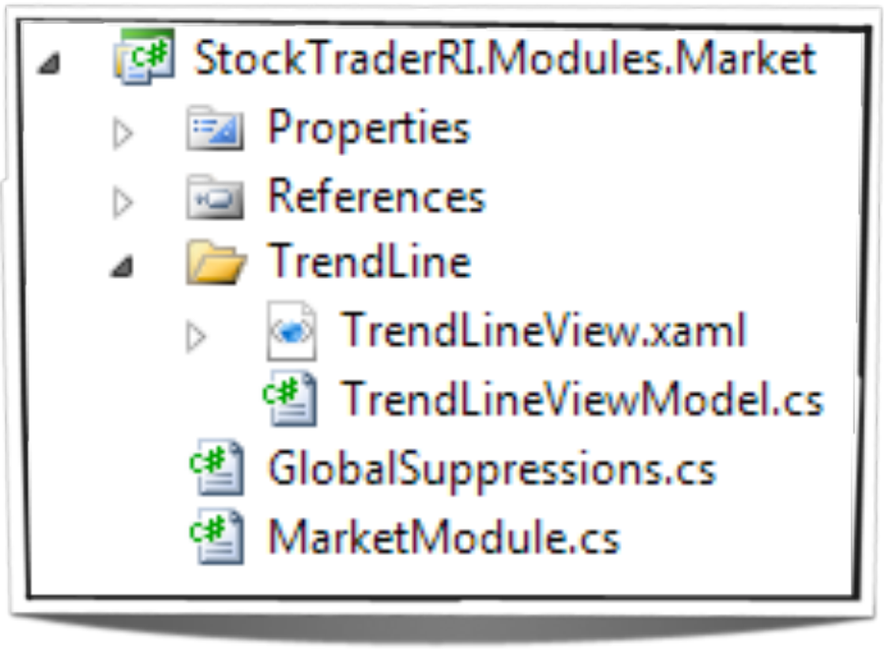
```
// views/personView.html
<div>My name is: <span class="name"></span></div>

// views/personView.js
define(['text!./personView.html'], function(tmpl) {
    function PersonView(options) {
        var el = $(options.el),
            name = options.name;

        el.html(tmpl).find('.name').text(name);
    }
    return PersonView;
});
```

Ask for content with the text plugin and we get a `String`

```
// views/personView.html
<div>My name is: <span class="name"></span></div>

// views/personView.js
define(['text!./personView.html'], function(tmpl) {
    function PersonView(options) {
        var el = $(options.el),
            name = options.name;

        el.html(tmpl).find('.name').text(name);
    }
    return PersonView;
});

// index.html
<div id="person"></div>
<script>
    require(['views/personView'], function(PersonView) {
        var personView = new PersonView({
            el: $('#person'), name: 'David'
        });
    });
</script>
```

Ask for content with the text plugin and we get a `String`

This allows us to organize our applications the way we're used to



C#

JavaScript

(Yep, I know you would've guessed...)

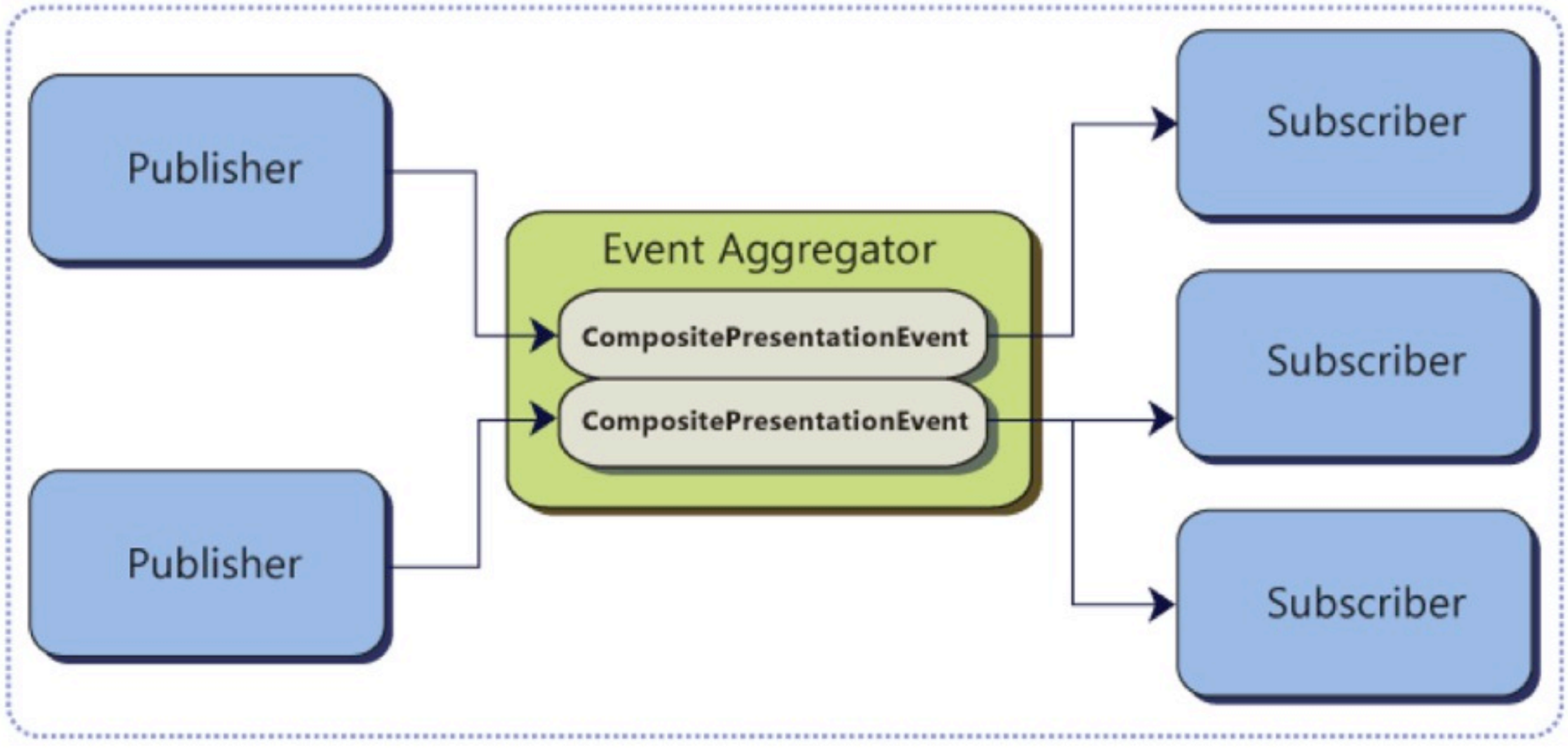
Demo

"Controls" with RequireJS

Event Aggregator

The **EventAggregator** service is primarily a container for events that allow decoupling of publishers and subscribers so they can evolve independently. This decoupling is useful in modularized applications because new modules can be added that respond to events defined by the shell or, more likely, other modules.

In the Composite Application Library, **EventAggregator** allows subscribers or publishers to locate a specific **EventBase**. The event aggregator also allows for multiple publishers and multiple subscribers, as shown in Figure 1.



```
require(['pubsub'], function(PubSub) {
    var bus = new PubSub();

    bus.sub('say', function(msg) {
        console.log('Subscriber 1 says: ' + msg);
    });

    bus.sub('say', function(msg) {
        console.log('Subscriber 2 says: ' + msg);
    });

    bus.pub('say', 'Nope!');
    // Subscriber 1 says: Nope!
    // Subscriber 2 says: Nope!
});
```

```
define('pubsub', function() {
    function PubSub() {
        this.subs = {};
    }

    PubSub.prototype = {
        sub: function(topic, fn) {
            var topicSubs = (this.subs[topic] = this.subs[topic] || []);
            topicSubs.push(fn);
        },
        pub: function(topic, data) {
            var topicSubs = this.subs[topic] || [];
            topicSubs.forEach(function(fn) {
                fn(data);
            });
        }
    };
    return PubSub;
});
```

"Dictionary" to hold topic to list of subscriptions

Create topic list if none exists

Call each subscriber for topic

(This is is a horribly crap implementation - DON'T EVER USE IT!)

Demo

pubsub

The ease of Unit Testing is one of the biggest plus points for JavaScript

There's a lot less fighting the compiler.
Which is handy, as they're useful as there is no compiler.

There is a large number of unit testing libraries and frameworks

QUnit (of jQuery fame)
Jasmine (BDD'ish)

```
describe('Calculator', function() {
    var calculator;

    beforeEach(function() {
        calculator = require('calculator');
    });

    it('should add two numbers correctly', function() {
        expect(calculator.add(2, 3)).toEqual(5);
    });
});
```

Jasmine 1.0.2 revision 1298837858 Show passed skipped

124 specs, 0 failures in 0.236s Finished at Fri Aug 05 2011 12:16:07 GMT-0400 (Eastern Daylight Time) run all

Demo

Unit Testing jQuery and mocks

CSS is wonderfully simple.

```
selector {
    property: value;
}
```

But sometimes it feels a little too simple.

Take it up a level - CSS Compilers

SASS

WebWorkbench (VS), SassAndCoffee (ASP.NET), Ruby

LESS

WebWorkbench (VS), dotless (ASP.NET), Java, Ruby, Node, JavaScript

Stylus

Node

Repetition in CSS sucks

```
.content-navigation {
  border-color: #3bbfce;
}

.border {
  border-color: #3bbfce;
}
```

SASS gives us variables

```sass
$blue: #3bbfce;

.content-navigation {
  border-color: $blue;
}

.border {
  border-color: $blue;
}
```

SASS

```css
.content-navigation {
  border-color: #3bbfce;
}

.border {
  border-color: #3bbfce;
}
```

CSS

Having to repeat selector paths sucks

```
header { background-color: blue; }

header a { font-weight: normal; }

header a:hover { font-weight: bold; }
```

SASS gives us nesting

SASS

```
header {
    background-color: blue;

    a {
        font-weight: normal;
        &:hover {
            font-weight: bold;
        }
    }
}
```

CSS

```
header { background-color: blue; }

header a { font-weight: normal; }

header a:hover { font-weight: bold; }
```

Having to repeat vendor prefixes sucks

```
#container {
    -ms-box-flex: 1;
    -o-box-flex: 1;
    -webkit-box-flex: 1;
    -moz-box-flex: 1;
    box-flex: 1;
}
```

SASS gives us mixins

```
@mixin box-flex($flex) {
    -ms-box-flex: $flex;
    -o-box-flex: $flex;
    -webkit-box-flex: $flex;
    -ms-box-flex: $flex;
    box-flex: $flex;
}

#container {
    @include box-flex(1);
}
```

SASS

```
#container {
    -ms-box-flex: 1;
    -o-box-flex: 1;
    -webkit-box-flex: 1;
    -moz-box-flex: 1;
    box-flex: 1;
}
```

CSS

Platform Consistency

Or, everyone else's browser sucks.

Writing an app for a single browser is easy.

Writing a single app for multiple browsers can be hard.

Just ask it

```
function hasCssProp(prop) {
    var el = document.createElement('div'),
        style = el.style;

    return typeof style[prop] == 'string';
}

// In IE9, Chrome, Firefox, etc...
hasCssProp('borderRadius'); // true

// In IE6, IE7, IE8, etc...
hasCssProp('borderRadius'); // false
```

"An indispensable tool."
— Bruce Bowman, , *Adobe BrowserLab Product Manager*

Modernizr is an open-source JavaScript library that helps you build the next generation of HTML5 and CSS3-powered websites.

Why use Modernizr?

Taking advantage of the new capabilities of HTML5 and CSS3 can mean sacrificing control over the experience in older browsers. Modernizr 2 is your starting point for making the best websites and applications that work exactly right no matter what browser or device your visitors use.

Thanks to the new Media Query tests and built-in YepNope.js micro-library as `Modernizr.load()`, you can now combine feature detection with media queries and conditional resource loading. That gives you the power and flexibility to optimize for every circumstance.

Check out the full list of features that Modernizr detects, or learn more about conditional resource loading with Modernizr.

Download Modernizr 2

View documentation

Use the commented, uncompressed Development version to develop with and learn from.

DEVELOPMENT
Uncompressed, 42 Kb

Then, dive into the Production build tool and pick just the tests you need!

PRODUCTION
Configure Your Build

Get started with Modernizr

While Modernizr gives you finer control over the experience through JavaScript-driven feature detection, it is important to continue to use best practices throughout your development process. Use progressive enhancement wherever you can, and don't sacrifice accessibility for convenience or performance.

- Documentation: Getting started
- Taking Advantage of HTML5 and CSS3 with Modernizr, *Faruk Ateş*
- How to use Modernizr, *Inayaili de León*
- [wiki] The Undetectables: features that cannot be detected
- [wiki] Cross-browser Polyfills

Also check out our Resources section.

Tip: use haz.io to quickly test your current browser's features.


```
<!DOCTYPE html>
▼<html class=" js flexbox canvas canvastext webgl no-touch
 geolocation postmessage websqldatabase indexeddb hashchange
 history draganddrop websockets rgba hsla multiplebgs
 backgroundsize borderimage borderradius boxshadow textshadow
 opacity cssanimations csscolumns cssgradients cssreflections
 csstransforms csstransforms3d csstransitions fontface
 generatedcontent video audio localstorage sessionstorage
 webworkers applicationcache svg inlinesvg smil svgclippaths">
 ▶<head>…</head>
```


```
<!-- DOCTYPE html -->
<html class=" js no-flexbox no-canvas no-canvastext no-webgl no-touch no-geolocation
        postmessage no-websqldatabase no-indexeddb hashchange no-history draganddrop
        no-websockets no-rgba no-hsla no-multiplebgs no-backgroundsize no-borderimage
```

Demo

Modernizr + SASS + CSS3 Hawtness + CSS2 Not-so-hawtness

The elephant in the room...

“But my company is all IE6...”

progressive internet explorer

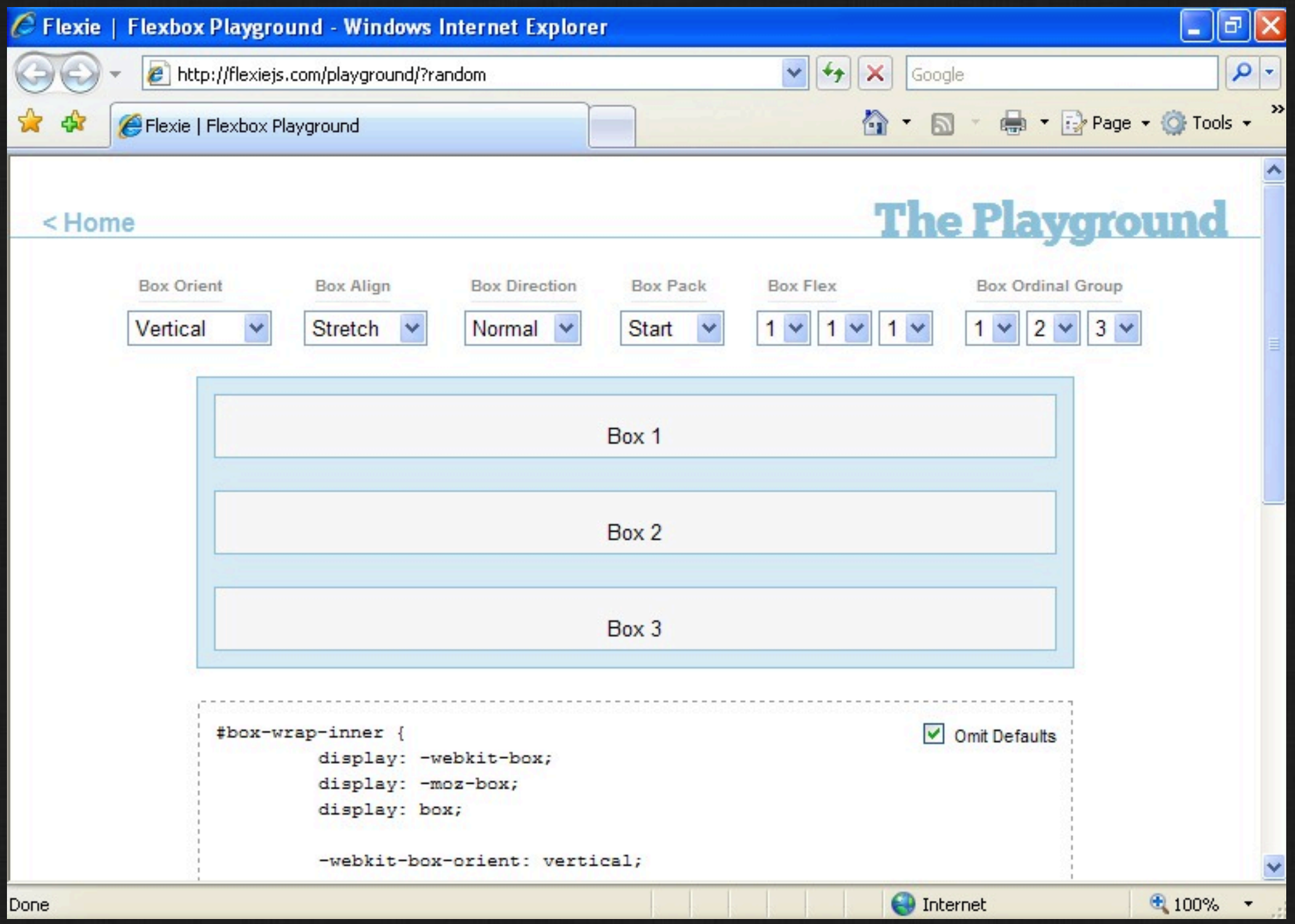
CSS3 PIE

```
border-radius: 8px;
box-shadow: #666 0px 2px 3px;
background: linear-gradient(#eeff99, #66ee33);
behavior: url(/PIE.htc);
```

http://css3pie.com/

Flexie

Cross-browser support for the Flexible Box Model



You can fake a lot.

But honestly, it sucks.

Accept reality.

Chrome Frame - a plugin with benefits

System Admin Compatible

Roll out with MSI

Control with Group Policy Settings

User Installable

Just include a script in page

Non-admin installs supported

Keep IE

Sites have to explicitly opt-in to Chrome Frame

Sites depending on IE6/7 stay in IE6/7

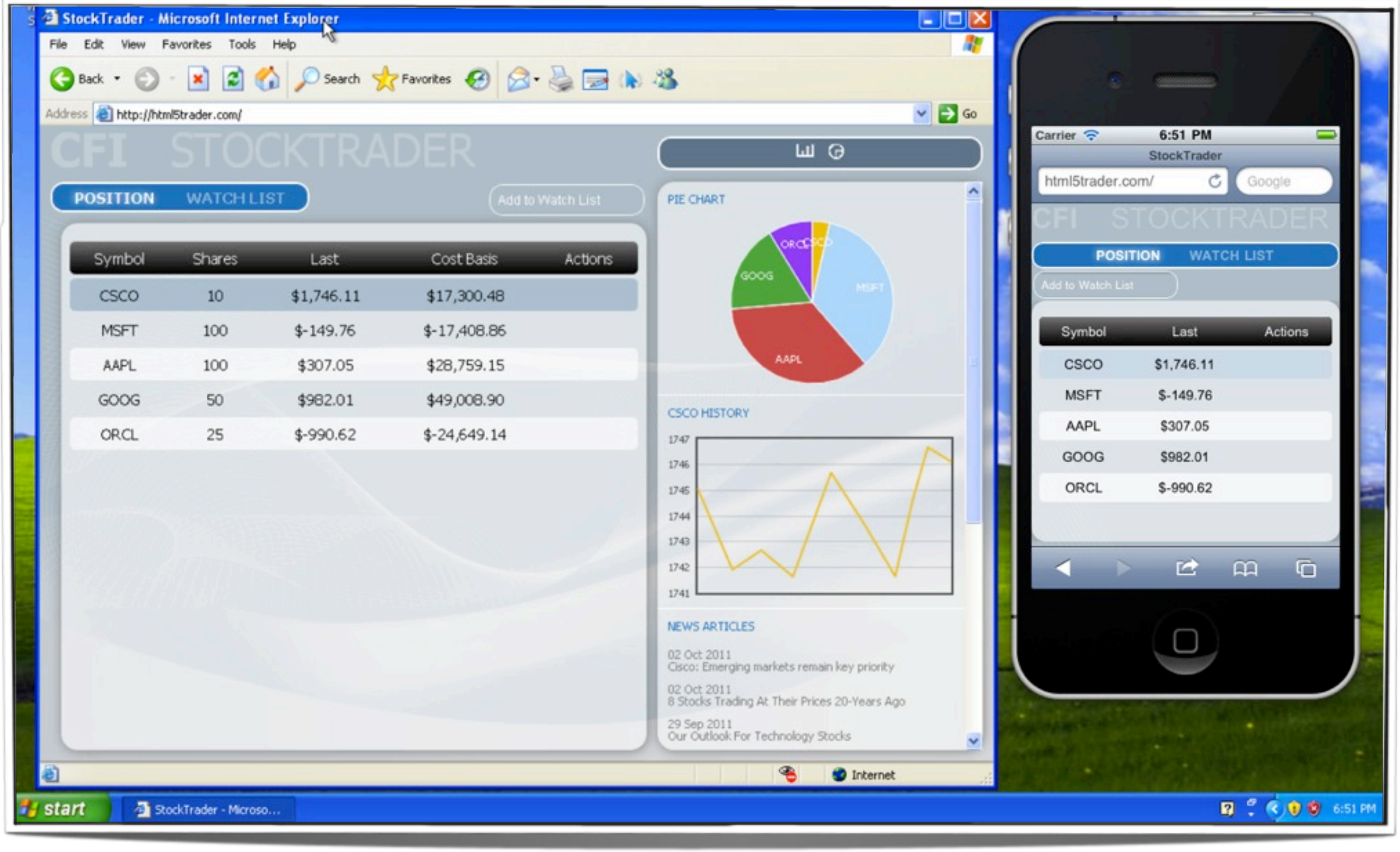
http://www.google.com/chromeframe

```
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
```

Applications have to explicitly opt-in so your legacy applications won't be affected

Demo

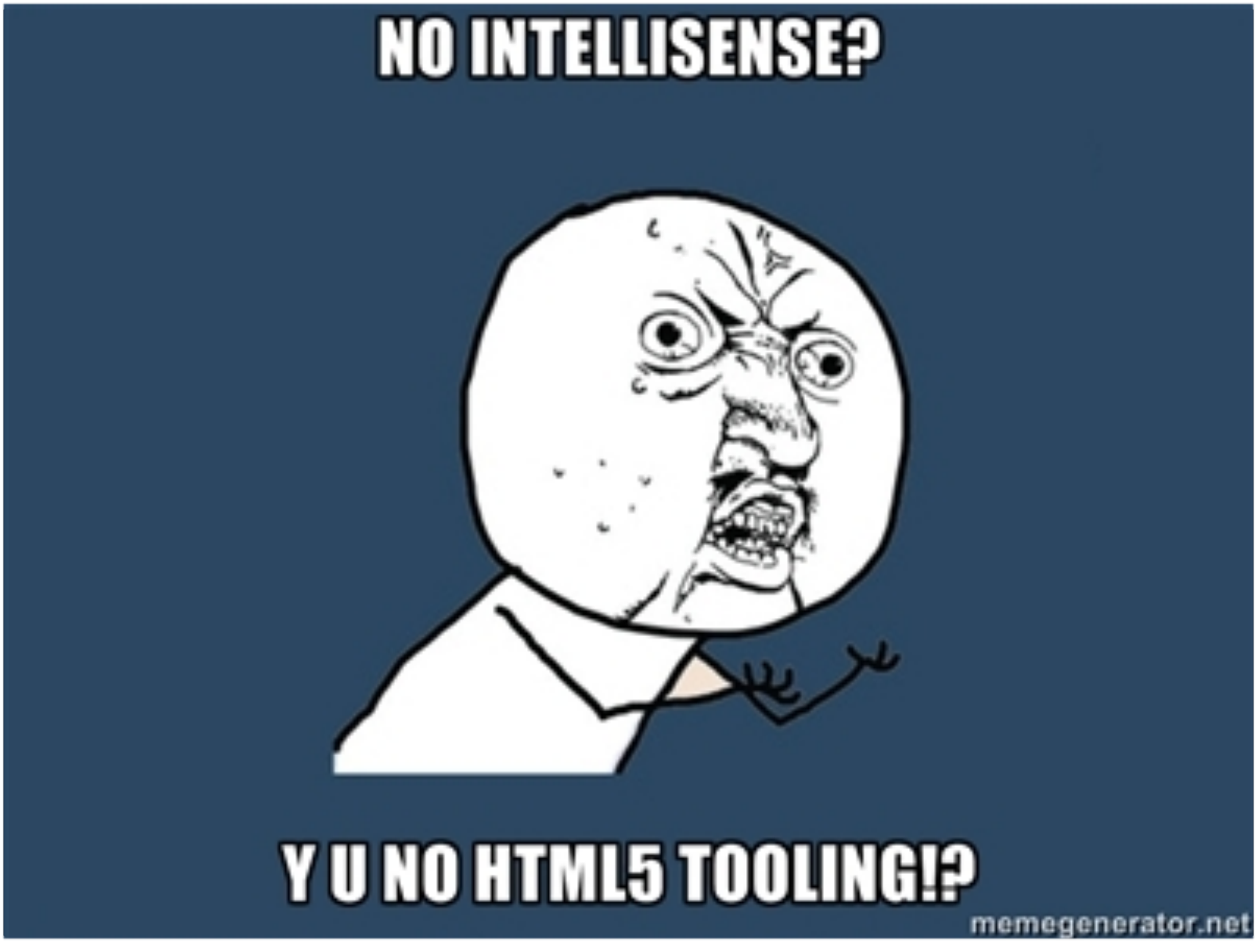
HTML5 Everywhere



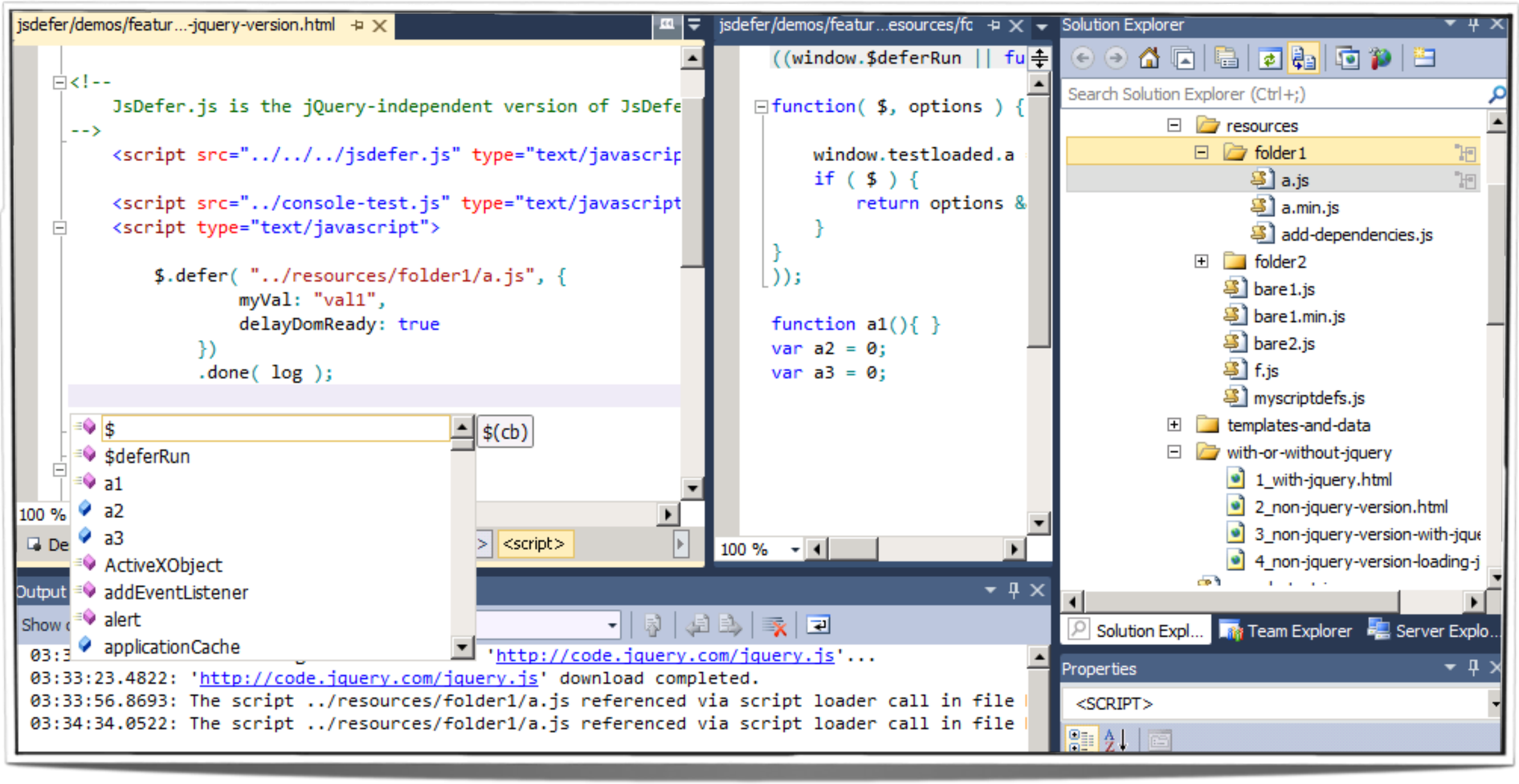
Running the same application - http://html5trader.com

Running the same application - http://html5trader.com

“This HTML5 stuff would be fine, if only there was decent tooling...”


NO INTELLISENSE?
Y U NO HTML5 TOOLING!?
memegenerator.net

Visual Studio 11 is looking much better...



http://blogs.msdn.com/b/webdevtools/archive/2011/09/15/new-javascript-editing-features-for-web-development-in-visual-studio-11-developer-preview.aspx

WebStorm 2.1

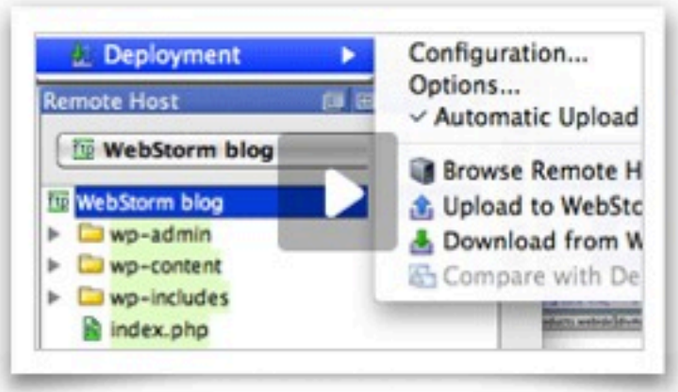
WebStorm — The smartest JavaScript IDE

Create great web sites in a great IDE

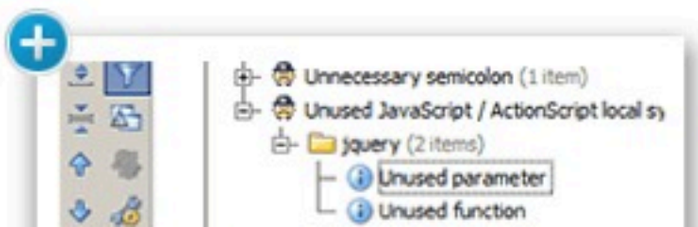
The best JavaScript IDE with HTML editor is at your fingertips. Navigate through files easily. Use relevant autocompletion for everything in your code. Get notified about code problems on the fly. Complicated languages mixtures with HTML markup or SQL inside a JavaScript? Check how a modern IDE such as WebStorm handles this.

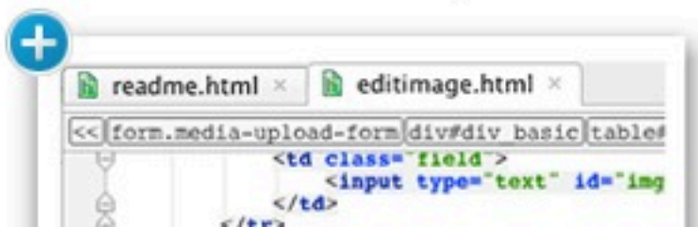
Open code from anywhere and start working in no time

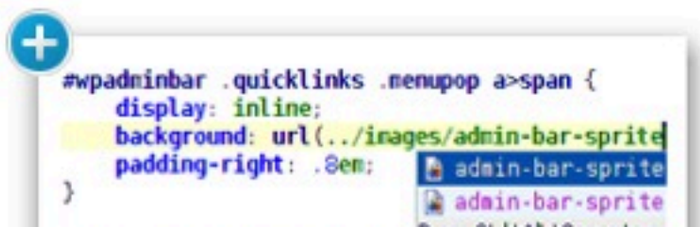
Open an existing folder, check out the code from a VCS, or even specify your FTP to download and auto-sync your files with. You're up and running in moments. Watch this short demo on getting started.

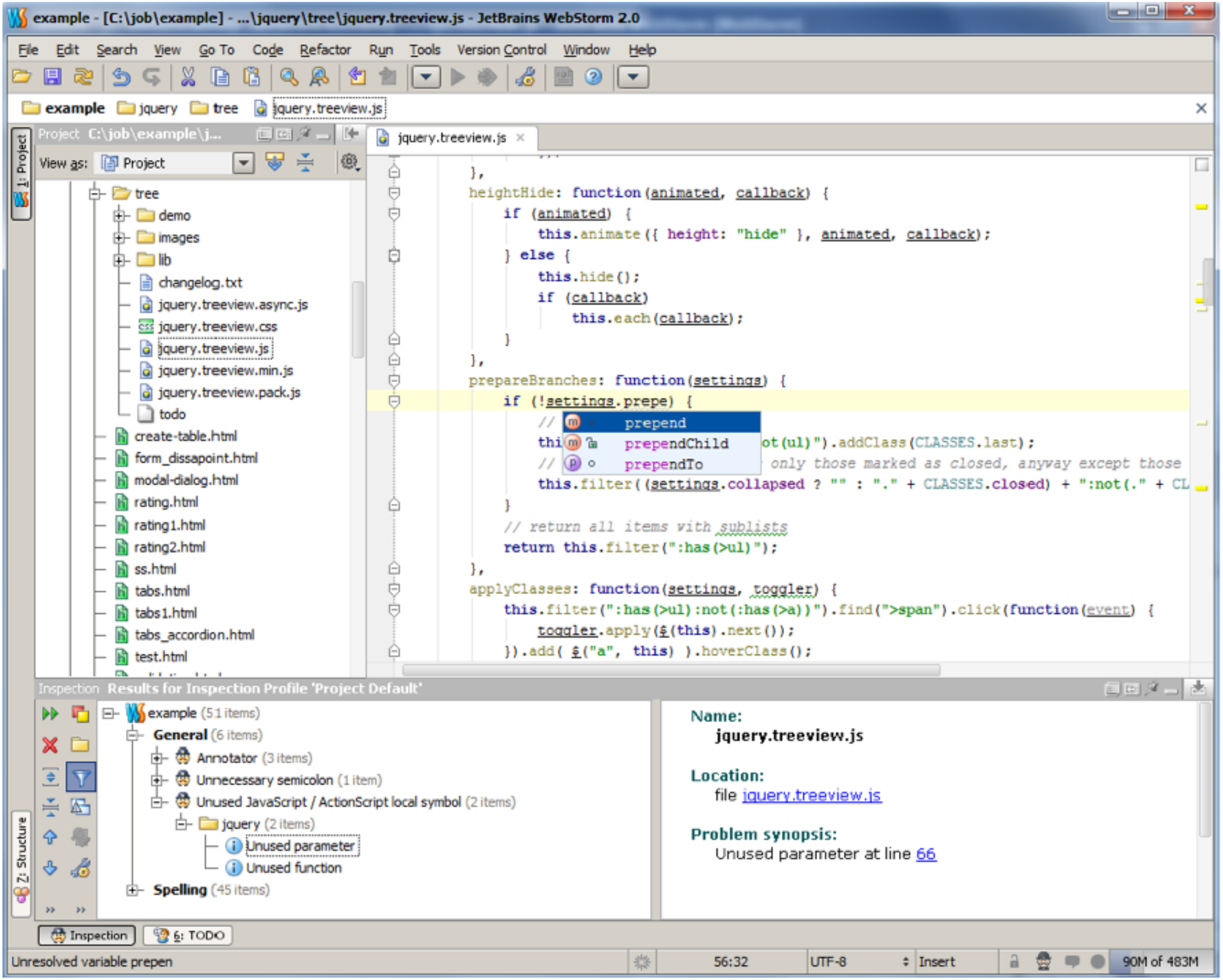


Use the same environment on Windows, Mac OS and Linux



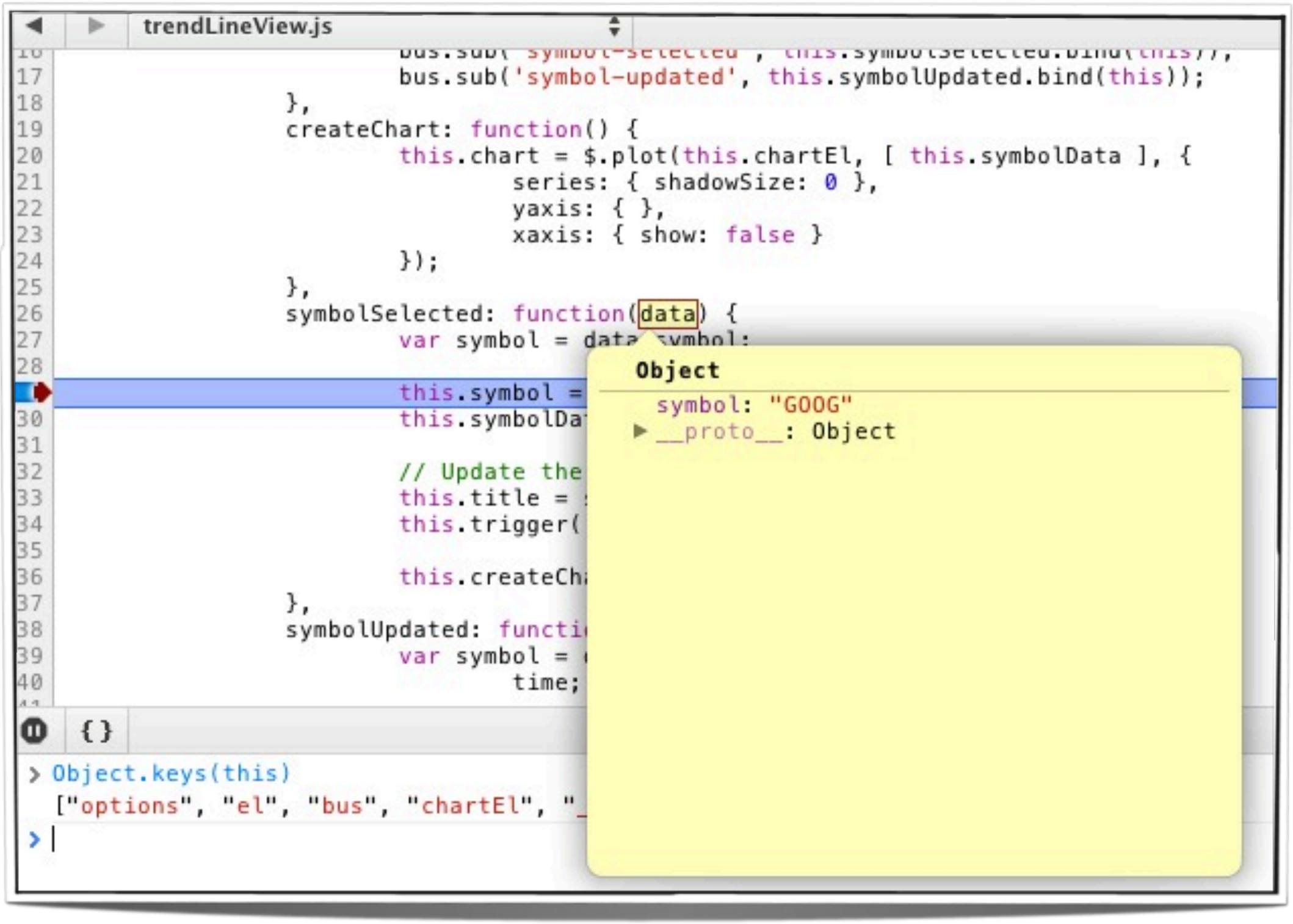





example - [C:\job\example] - ...\jquery\tree\jquery.treeview.js - JetBrains WebStorm 2.0
File Edit Search View Go To Code Refactor Run Tools Version Control Window Help
example jquery tree jquery.treeview.js
Project C:\job\example\j...
View as: Project
tree
demo
images
lib
changelog.txt
jquery.treeview.async.js
jquery.treeview.css
jquery.treeview.js
jquery.treeview.min.js
jquery.treeview.pack.js
todo
create-table.html
form_dissapoint.html
modal-dialog.html
rating.html
rating1.html
rating2.html
ss.html
tabs.html
tabs1.html
tabs_accordion.html
test.html
jquery.treeview.js
},
heightHide: function(animated, callback) {
if (animated) {
this.animate({ height: "hide" }, animated, callback);
} else {
this.hide();
if (callback)
this.each(callback);
}
},
prepareBranches: function(settings) {
if (!settings.prepe) {
// prepend
thi prependChild ot(ul)").addClass(CLASSES.last);
// prependTo only those marked as closed, anyway except those
this.filter((settings.collapsed ? "" : "." + CLASSES.closed) + ":not(." + CL
}
// return all items with sublists
return this.filter(":has(>ul)");
},
applyClasses: function(settings, toggler) {
this.filter(":has(>ul):not(:has(>a))").find(">span").click(function(event) {
toggler.apply($(this).next());
}).add($("a", this)).hoverClass();
Inspection Results for Inspection Profile 'Project Default'
example (51 items)
General (6 items)
Annotator (3 items)
Unnecessary semicolon (1 item)
Unused JavaScript / ActionScript local symbol (2 items)
jquery (2 items)
Unused parameter
Unused function
Spelling (45 items)
Name:
jquery.treeview.js
Location:
file jquery.treeview.js
Problem synopsis:
Unused parameter at line 66
Inspection
6: TODO
Unresolved variable prepen
56:32
UTF-8
Insert
90M of 483M

We use Intellisense to ‘Explore’ and ‘Discover’

http://www.flickr.com/photos/gemmabou/4622213880/



With modern browsers we can easily and quickly explore our code at runtime

Smaller Surfaces

```
// Getter
$('#test').css('backgroundColor');

// Setter
$('#test').css('backgroundColor', 'red');
```

```
// Bigger setter
$('#test').css({
    backgroundColor: 'red'
});
```

Static analysis for JavaScript

```
function add(a, b, c) {
  return a + b;
}
```

```
function add(a, b, c) {
  return a + b;
}
```

Error:

Unused variable: c *1* add

```
var person = {
    name: 'David',
    company: 'Lab49',
};
```

```
var person = {
    name: 'David',
    company: 'Lab49',
};
```

Error:

Problem at line 3 character 25: Unexpected ','.

```
company: 'Lab49',
```

```
function createPerson(name) {
    return
    {
        name: name
    };
}
```

```
function createPerson(name) {
    return
    {
        name: name
    };
}
```

Error:

Problem at line 2 character 11: Expected ';' and instead saw '{'.

```
return
```

Problem at line 3 character 5: Unreachable '{' after 'return'.

```
{
```

There's tons of other tools....

Automation Testing
Validators
Performance
Debuggers
Profilers
Designers

On many other platforms...

.NET
Java
Ruby
Python
Node

It is not Silverlight vs HTML5

They’re both valid options with different pros and cons

You guys are experts in building front-end applications.

Although the code looks a little different...

The patterns, techniques and good practices are the same.

Be Confident

Be Fearless

(and be sensible)

Hang on,

You went an entire talk to a Silverlight audience and didn't once mention MVVM?

Yep - although you can reuse a lot of knowledge, don't try to do everything exactly the same.
I encourage you to start with the simplest thing and build from there.

You didn't cover ...!?

Acceptance Testing

CoffeeScript

Underscore

Templates

Server Tools

IOC

jQuery

JavaScriptMVC

KnockOut

Flow Control

HTML5 Boiler Plate

Project Silk

BDD

Backbone

ECMAScript 6

ASP.NET Integration

jQuery UI

Promises

Goldfish

Script Loaders

Dec 6th

RequireJS 1.0

Callbacks, Promises and Coroutines (on my!)

The NY HTML5 Application Developers Group

Dec 6th

RequireJS 1.0

Callbacks, Promises and Coroutines (on my!)

http://www.meetup.com/html5-app-developers

New York .NET Meetup

http://www.meetup.com/NY-Dotnet

http://www.lab49.com

```
// Thanks for listening!

return;
```